MetaCentrum

Distributed job scheduling in MetaCentrum – The Czech NGI

Šimon Tóth Miroslav Ruda

CESNET z.s.p.o.

September 4, 2014

Czech National Grid

Core parameters

- 10 000 CPU cores
  550 nodes, 12GB..6TB RAM per node
- 1 PB permanent storage
- 27 PB long-term storage

Talk overview

- quick introduction to job scheduling
- virtualized infrastructure
- distributed scheduling
- fairness

Job scheduling

- input:
  - user submitted jobs with resource req. and constraints
- output:
  - assignment of jobs to physical resources in time and space
- NP complete problem
- fuzzy definition optimality for production systems

Torque Batch System

- 5 years of usage from PBSPro switch
- locally maintained fork based on 2.4 stable branch
  `https://cesnet.github.io/torque/`
- custom scheduler

Primary use cases

- workloads that are not feasible on desktop machines
  long jobs, large submissions, parallel jobs
- best-effort model based on fairness

Resource management

- resources exclusively allocated to users
- standard resources

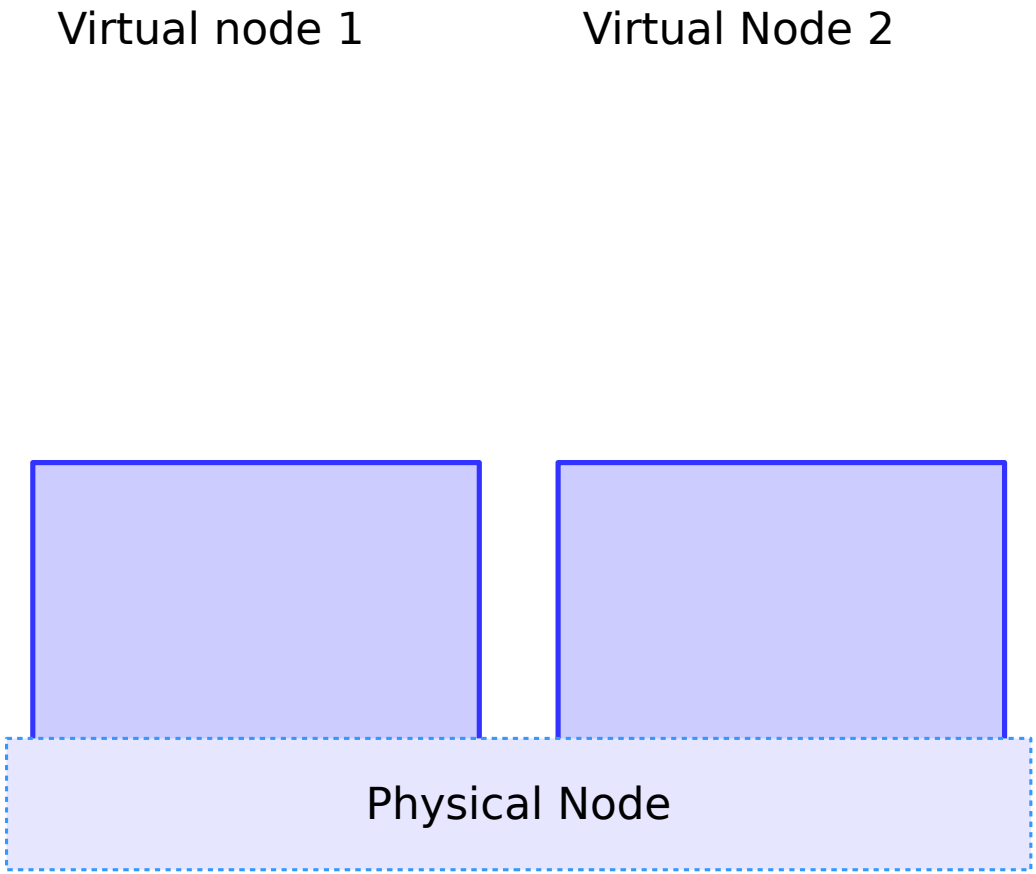
  CPU cores, Memory, GPU cards
- resources with uncertain states

  software Licenses, scratch disk space

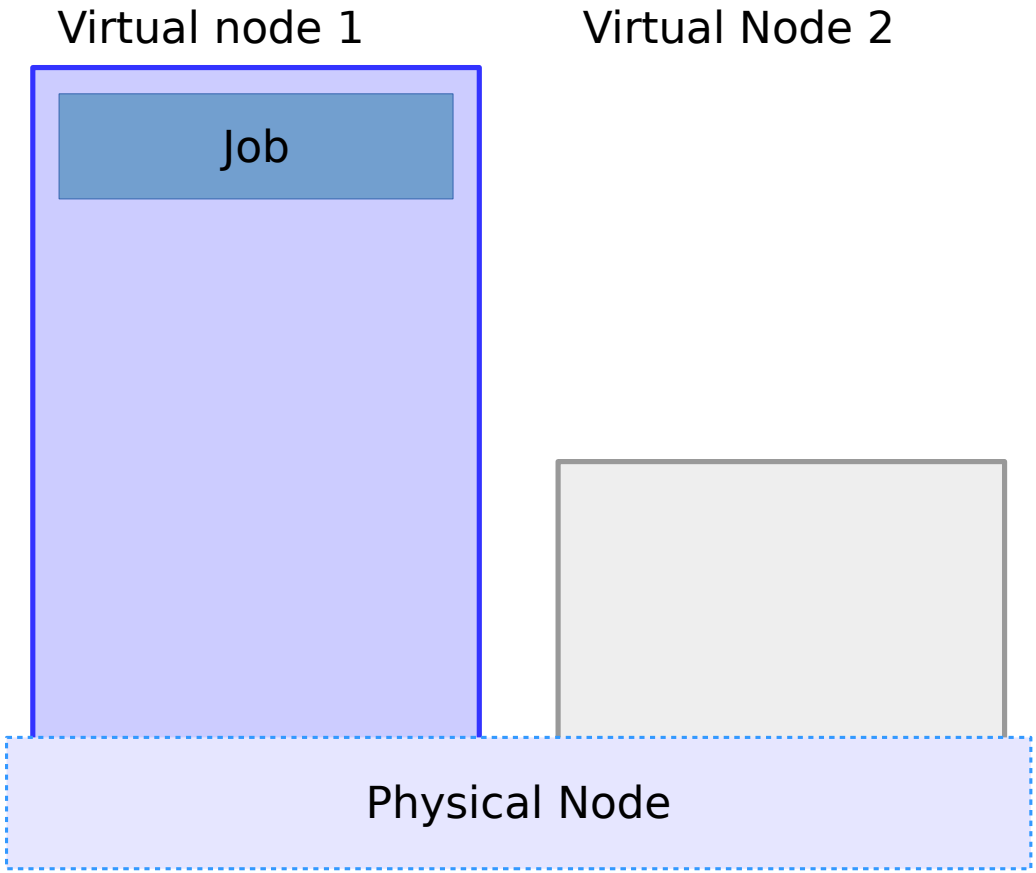
Full machine preemption

- two virtual machines switching resources
- reasonably fast (minutes) switch time
- allows full-machine preemption

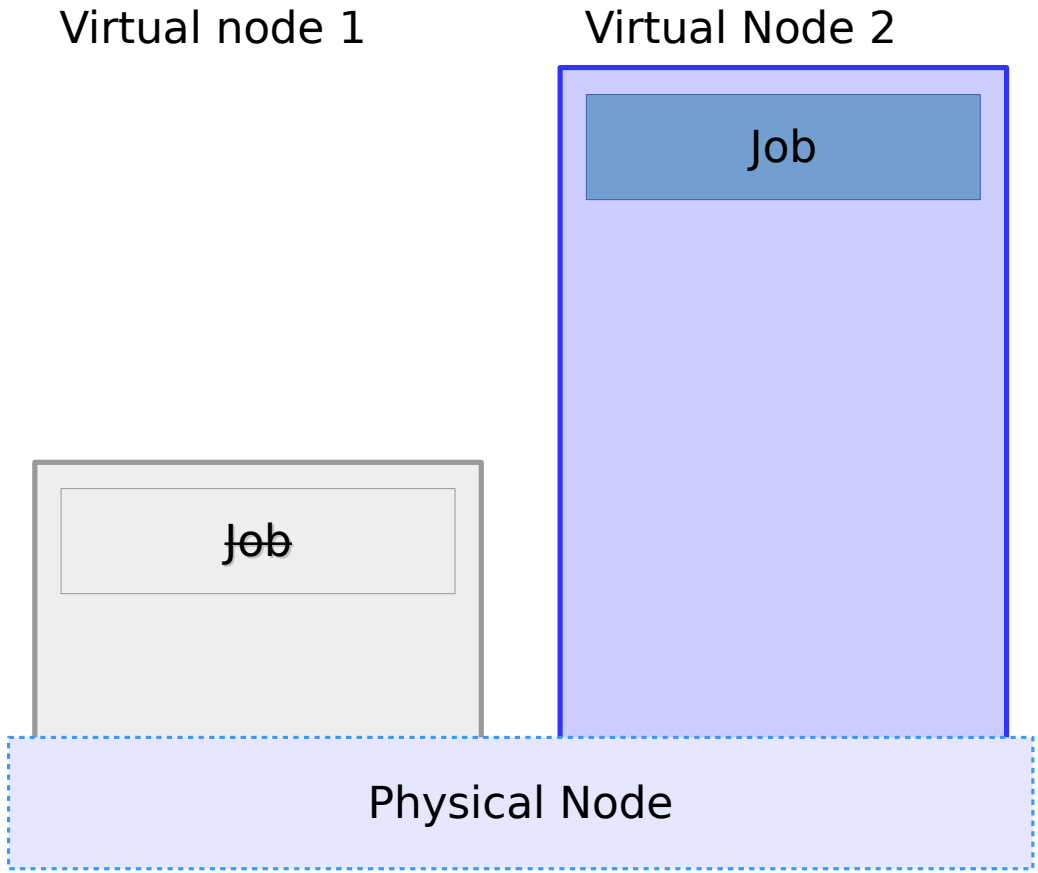
Full machine preemption



Full machine preemption



Full machine preemption



Ondemand virtual machines

- improves variety of supported software configurations
- scheduler invokes machine reboots
- virtual image selected using requested software properties

Virtual clusters

- user requested virtual clusters
- possibility to use VLAN
- clusters can connect back to resource manager
- or fully under control of user

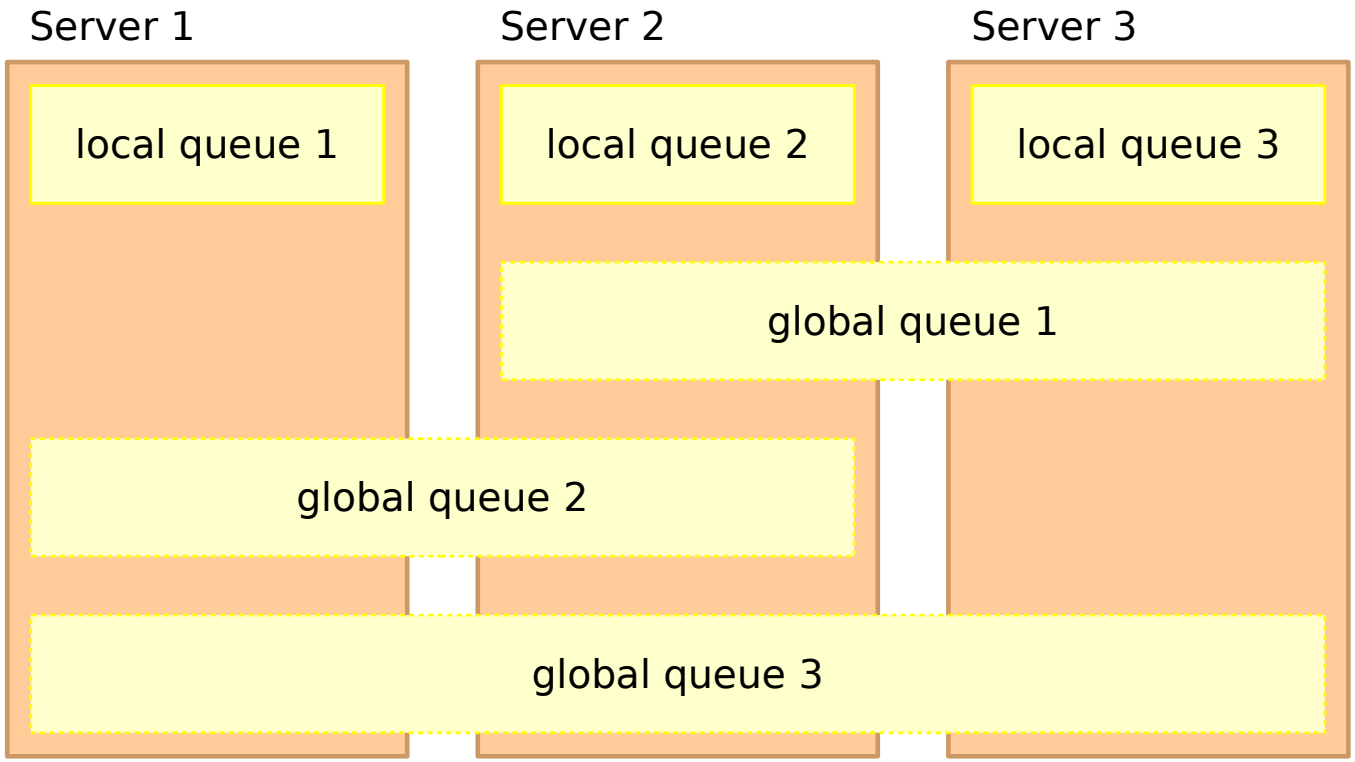
P2P scheduling

- improve resilience
- increase performance
- allow separate sites to maintain local policies

P2P scheduling

- each scheduler responsible for its local resources
- all jobs behave as if submitted locally
- running job information synchronized between servers

P2P scheduling



P2P scheduling

- very low demands on sites
- each site maintains complete autonomy
- outages heavily localized

Fairness

- long-term fairness based on resource consumption
- users with high publication count receive higher share
- implemented using fairshare ordering policy

Fairness

- virtual fairshare queue
- multi-resource fairness

Conclusion - other features

- Kerberos authentication
- scheduling of advanced resources

  GPU, software licenses, scratch space
- complex job specification

  resources, properties, negative requests, location requests
- enhanced queue semantics

  resource limits, node access control

Summary

- virtualized infrastructure

  full-machine preemption, ondemand machines, virtual clusters
- distributed scheduling
- fairness model

  fairshare, multi-resource fairness

Further Reading

M. Ruda et al.
Virtual Clusters as a New Service of MetaCentrum, the Czech NGI.
Cracow Grid Workshop'09, 2009.

Š. Tóth, and D. Klusáček.
User-Aware Metrics for Measuring Quality of Parallel Job Schedules.
Workshop on Job Scheduling Strategies for Parallel Processing, 2014.

D. Klusáček, and H. Rudová.
Multi-Resource Aware Fairsharing for Heterogeneous Systems.
Workshop on Job Scheduling Strategies for Parallel Processing, 2014.

D. Klusáček, and Š. Tóth.
On Interactions among Scheduling Policies: Finding Efficient Queue Setup Using High-Resolution Simulations.
Euro-Par 2014 Parallel Processing, 2014.